

Remarks

Applicant has amended claims 1, 2, 7, 18-22, 25-27 and 31. Applicant respectfully submits that no new matter was added by the amendment, as all of the amended matter was either previously illustrated or described in the drawings, written specification and/or claims of the present application. Entry of the amendment and favorable consideration thereof is earnestly requested.

The Examiner has rejected claims 14, 2, 4, 7-10 and 15 under 35 USC 102(b) as being anticipated by International Application Published Under The PCT No. WO 02/093371 (Laitila). Claims 5, 6, 11-13, 25-33, 43-45, 47 and 48 are rejected under 35 USC 103(a) as being unpatentable over Laitila. Applicant respectfully disagrees.

Claim 14 recites “an emulated Application Programming Interface library having a table and accessible by said computer, said library including data indicative of types of data manipulations between the first computer language source code and the second computer language source code, said table including second computer language equivalent functions callable by said second computer language.” Claim 43 recites “an emulation Application Programming Interface library having a table and accessible by said computer including data indicative of types of data manipulations between the Java source code and the OOP language source code, said table including OOP language equivalent functions callable by said OOP computer language.”

Claim 25 recites “referencing an emulation Application Programming Interface library” and “the second computer language source code referencing the Application Programming Interface library based on the selected equivalent functions emulating the first computer language.” Claims 47 and 48 recite “referencing an emulation Application Programming Interface library” and “the second OOP computer language referencing the Application Programming Interface library to perform a data manipulation with the

second OOP computer language source code that emulates the type of data manipulation the first OOP computer language source code performs.”

Initially, Applicant would like to point out that Laitila discloses two distinct methods, a “method for developing a translator” and use of the “corresponding system” or the developed translator itself. (p. 1, Ins. 3-4; p. 2, Ins. 14-15). Therefore, a distinction needs to be made between functions relating to the method for developing a translator and the method of actually translating from one programming language to another.

Applicant respectfully submits that Laitila does not disclose or teach an “Application Programming Interface library” as recited in all the pending claims.

In addition, Applicant respectfully submits that Laitila does not disclose or teach “second computer language equivalent functions callable by said second computer language” as recited in claims 14 and 43. Laitila also fails to disclose or teach “the second computer language source code referencing the Application Programming Interface library based on the selected equivalent functions emulating the first computer language” as recited in claim 25, or “the second OOP computer language referencing the Application Programming Interface library to perform a data manipulation with the second OOP computer language source code that emulates the type of data manipulation the first OOP computer language source code performs” as recited in claims 47 and 48.

Applicant respectfully submits that the term Application Programming Interface (API) is a set of functions or “classes” for working with a virtual machine, a runtime environment, or an operating system that a “program may call.” (See, para. 41) The term emulated API library comprises a set of functions or “classes” implemented in a second computer environment that work just like a corresponding set of functions or “classes” that are commonly provided for working with a first computing environment, such as a virtual machine, a runtime environment, or an operating system, that a “program [in the second computing environment] may call.” (See, para. 41) Laitila, however, does not

disclose or teach use of an emulated API library. At most, Laitila discloses an “operations library”, which does not implement any API functions or API classes and cannot serve as an emulated API library for any general-purpose computing environment. Rather, the “operations library” comprises functions that are called directly by the translator itself, not by any translated source code. [See, p. 6, ln. 16] Nowhere does Laitila disclose or teach that the “operations library” is referenced in the source code of the translated programs or that it is even accessible by the translated programs.

To clarify, with regard to the presently pending claims, Applicant respectfully submits that the pending specification teaches that the “system may be utilized as either an automatic translator...or may be utilized in connection with a *software development library* that enables programs to be manually ported from one computer language to another.” (Para. 0028) (emphasis added). “Application programming” and “software development” are synonymous terms, and an “emulated API library” is provided for the purpose of software development so a “software development library” may also be called an “API library.” The specification further teaches that “a *software development library* that *implements* the Java language and core *APIs* in ANSI standard C++ syntax is provided. Manually ported C++ *programs*, which *make use of the library*, closely resemble the original programs in the Java language.” (Para. 0030) The foregoing statement also fully applies to automatically translated programs. This shows that even if there was no translator program involved and instead a human was making the translation manually (which is called a “port”), the translated or “ported C++ programs” would still “make use of the [API or software development] library.” This is very different from Laitila where the “operation library” is only used or called by the translator not by the translated programs themselves.

Accordingly, Applicant respectfully submits that Laitila fails to disclose or teach 1) an emulated API library, or 2) that the second computer language references or calls the API library as variously recited in all of the pending claims.

It is respectfully submitted that claims 2, 4-15, 25-33, 43-45 and 47-48, all of the claims remaining in the application, are in order for allowance and early notice to that effect is respectfully requested.

Respectfully submitted,

July 31, 2007

/Steven B. Simonis/
Steven B. Simonis, Registration No. 54,449
Attorneys for Applicant
ST. ONGE STEWARD JOHNSTON & REENS LLC
986 Bedford Street
Stamford, CT 06905-5619
Tel. 203 324-6155